CHAPTER

12

Working with the Windows XP Registry

After reading this chapter and completing the exercises, you will be able to:

- Understand the function and structure of the Registry
- Describe the purpose of each of the five Registry keys and the hive files to which some of them map
- Use the Registry editor and various other Registry tools (including several from the *Microsoft Windows XP Professional Resource Kit*)
- Understand default Registry sizing techniques and limits on Registry size
- Understand the fault-tolerance mechanisms for the Registry
- Back up and restore the Registry

Windows XP is a complex operating system that relies upon a dynamic data structure to maintain its configuration and operational parameters. This structure is the hierarchical database known as the **Registry**, which contains most of the control and functional settings for Windows XP core elements, services, and native applications, as well as many Microsoft and third-party add-on software products. In this chapter, you learn about the Registry, its structure, tools to edit and manage it, and several values you may consider altering to improve or configure system operation.

Windows Registry Overview

The Registry provides Windows XP with a hierarchical database of information about a system's configuration. The Registry stores information essential to Windows XP itself, native applications, added services, and most add-on software products from Microsoft and third-party vendors. The information stored in the Registry is comparable to that stored in initialization files (.ini, .dat, .bat, .sys, and so on) in Windows 3.x or even Windows 95/98 (which used both the Registry and .ini files). For native 32-bit Windows applications, the Registry database takes the place of .ini files and stores all configuration information. The Registry is not a text file, such as win.ini or CONFIG.SYS, but is rather a multifaceted branch-like grouping of data.

Although most Windows XP Professional configuration can be performed using the Control Panel applets and the Administration Tools (in fact, changes made to system configurations through these tools are applied to the Registry database), some settings can be established or changed only by editing the Registry directly. To edit the Windows XP Registry, you must use the Registry editor, which is launched by executing Regedit. This tool is discussed in detail later in this chapter.

Caution

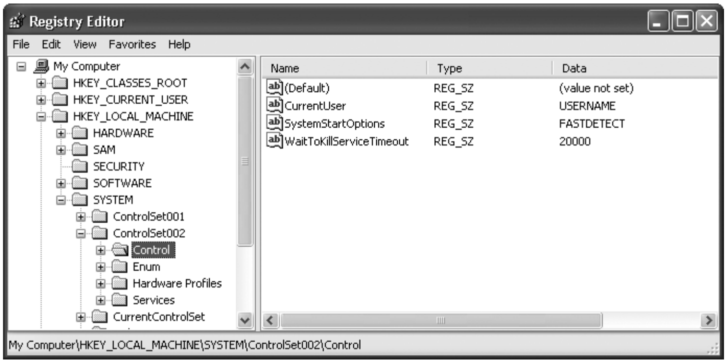
Microsoft warns that editing the Registry directly should only be performed when absolutely necessary. If possible, use Control Panel applets or Administrative Tools to make system modifications rather than manipulating values directly in the Registry. Improper editing of the Registry can cause system malfunctions and can even render the system completely inoperable.

The Registry was designed for programming ease and speed of interaction for processes. The Registry's structure, although a bit daunting, is understandable if broken down into its component parts. The Registry is divided into keys and subkeys. Each Registry **key** is similar to a bracketed heading in an .ini file and represents a top-level container in the Registry hierarchy. There are five of these highest-level, or root, keys; their names start with HKEY to designate their highest-level status. Each key may contain one or more lower-level keys called **subkeys**. Within each subkey, one or more values or subkeys can exist.

A **value entry** is a named parameter or placeholder for a control setting or configuration data. A value entry can hold a single binary digit, a long string of ASCII characters, or a hexadecimal value. The actual piece of data held by a value entry is known as the **value**. Figure 12-1 shows the structure of the Registry contents in Regedit. The left pane shows three of the five root keys, with subkeys displayed for the HKEY_LOCAL_MACHINE key. The right pane shows the value entries for the SYSTEM\ControlSet002\Control subkey.

Note

A discrete body of Registry keys, subkeys, and values stored in a file is also known as a **hive**. Such files reside in the *%systemroot%*\system32\WINDOWS\system32\config directory and normally correspond to some of the root keys shown in Figure 12-1 (e.g., the file named "system" corresponds to the HKEY_LOCAL_MACHINE\SYSTEM root key). For a complete listing of all hives on your system, use Regedit.exe to inspect the contents of the HKLM\SYSTEM\CurrentControlSet\Control\hivelist subkey.

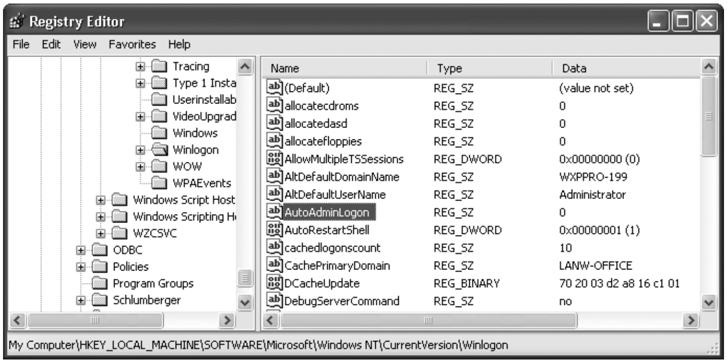


Name	Type	Data
(Default)	REG_SZ	(value not set)
CurrentUser	REG_SZ	USERNAME
SystemStartOptions	REG_SZ	FASTDETECT
WaitToKillServiceTimeout	REG_SZ	20000

My Computer\HKEY_LOCAL_MACHINE\SYSTEM\ControlSet002\Control

Figure 12-1 View of the hierarchical Registry structure, including five primary keys

Value entries within the Registry are composed of three parts: name, type, and data (value). A Registry value entry's name is typically a multiword phrase, without spaces, with title capitalization, such as AutoAdminLogon in Figure 12-2. The data type of a value entry informs the Registry how to store the value. The **data type** defines whether the piece of data is a text string or a number and gives the numerical base (radix) of that number. Radix types supported by Windows 2000 are decimal (base 10), hexadecimal (base 16), and binary (base 2). All hexadecimal values are listed with the prefix "0x" to identify them clearly (as in 0xF for 15).



Name	Type	Data
(Default)	REG_SZ	(value not set)
allocatecdroms	REG_SZ	0
allocatedasd	REG_SZ	0
allocatefloppies	REG_SZ	0
AllowMultipleTSSessions	REG_DWORD	0x00000000 (0)
AltDefaultDomainName	REG_SZ	WXPPRO-199
AltDefaultUserName	REG_SZ	Administrator
AutoAdminLogon	REG_SZ	0
AutoRestartShell	REG_DWORD	0x00000001 (1)
cachedlogonscount	REG_SZ	10
CachePrimaryDomain	REG_SZ	LANW-OFFICE
DCacheUpdate	REG_BINARY	70 20 03 d2 a8 16 c1 01
DebugServerCommand	REG_SZ	no

My Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon

Figure 12-2 AutoAdminLogon value entries

The data types supported by Windows 2000 are:

- ***Binary***—Binary format
- ***DWORD***—Binary, hex, or decimal format
- ***String***—Text-string format
- ***Multi-String***—Text-string format that contains multiple human-readable values separated by NULL characters
- ***Expandable String***—Expandable text-string format containing a variable that is replaced by an application when used (*%systemroot%\File.exe*)

Note

Once a value entry is created and its data type defined, that data type cannot be changed. To alter a value's data type, you must delete the value entry and re-create it with a new data type.

Important concepts to keep in mind about the Registry are:

- Keys are the top-level, or root, divisions of the Registry
- Keys contain one or more subkeys
- Any subkey can contain one or more subkeys
- Any subkey can contain one or more value entries

Also note that the Registry is not a complete collection of configuration settings. Instead, it holds only the exceptions to the defaults. Processes within Windows XP will operate with their own internal defaults unless a value in the Registry specifically alters that default behavior. This makes working with the Registry difficult: very often the control you need is not present in the Registry because internal defaults are in use. To alter such a setting, you'll need to add a new value entry to the Registry. To accomplish this, you must know the exact syntax, spelling, location, and valid values; otherwise, you will be unable to alter the default behavior. Keep in mind that failing to use the exact syntax, spelling, location, or valid values can result in malfunctions, possibly resulting in an inoperable system. So always edit with extreme care. The *Microsoft Windows XP Professional Resource Kit* includes a help file named Regentry.chm, which lists all possible Registry entries and valid values. This is an invaluable tool when attempting to modify existing Registry entries or when adding new ones.

Each time Windows XP boots, the Registry is loaded into memory from files (see "Registry Storage Files" later this chapter) stored on the hard drive. Each time Windows XP shuts down, the Registry is written from memory back to the files. While Windows XP is operating, the Registry remains in memory. This makes the Registry easy to access and quick to respond to control queries, and it is the reason why changes to the Registry take effect immediately. Only in extreme cases will Windows XP require a reboot to enforce changes in the Registry.

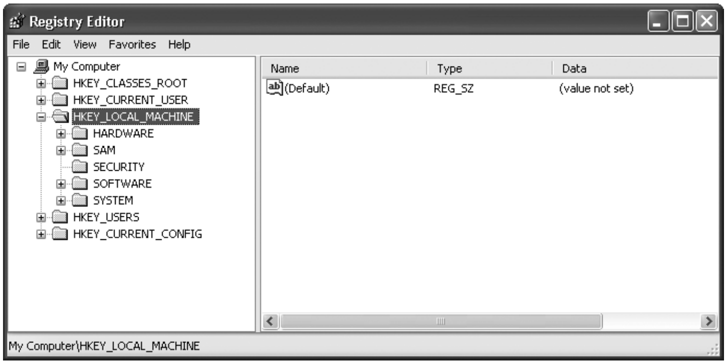
Important Registry Structures and Keys

In the following sections, we look at various keys and subkeys in the Registry and explain their functions.

HKEY_LOCAL_MACHINE

The **HKEY_LOCAL_MACHINE** key contains the value entries that control the local computer. These configuration items include information about hardware devices, applications, device drivers, kernel services, and physical settings. These data are used to establish the configuration of the hardware and operating system environment. The content of this key is not dependent on the logged-on user, or the applications or processes in use; it is dependent only on the physical composition of the hardware and software present on the local computer.

This key has five subkeys (see Figure 12-3): HARDWARE, SAM, SECURITY, SOFTWARE, and SYSTEM, which are described in the following sections.



Name	Type	Data
(Default)	REG_SZ	(value not set)

Figure 12-3 The HKEY_LOCAL_MACHINE key

HKEY_LOCAL_MACHINE\HARDWARE

The HKEY_LOCAL_MACHINE\HARDWARE subkey is the container for data related directly to physical devices installed on a computer. This subkey stores configuration data, device driver settings, mappings, linkages, relationships between kernel-mode and user-mode hardware calls, and IRQ hooks. This subkey is re-created each time the system boots and is not saved when the system shuts down. That explains why this subkey does not map to a specific hive file in the *%systemroot%*\WINDOWS\system32\config directory.

The HKEY_LOCAL_MACHINE\HARDWARE subkey contains three subkeys: DESCRIPTION, DEVICEMAP, and RESOURCEMAP. The DESCRIPTION subkey stores data extracted from a device's own firmware or onboard BIOS. The DEVICEMAP subkey stores information about device driver paths, locations, and filenames. The

RESOURCEMAP subkey stores information about the mappings between system resources (I/O ports, I/O memory addresses, interrupts, and direct memory access [DMA] channels) and device drivers. When certain bus types are present in the computer, a fourth subkey named OWNERMAP stores association information about the bus type and device drivers.

Note

The HKEY_LOCAL_MACHINE\Hardware subkey will contain a fourth subkey if your system contains software that supports the Advanced Configuration and Power Interface (ACPI). The ACPI subkey contains all of the operational parameters for that feature.

Caution

The contents of the HARDWARE subkey should not be manipulated. This key contains data read from the state of the physical devices and associated device drivers. There should be no need or reason to alter the data because they should always present a proper reflection of the state of the system. Second, these data are most often in binary format; so deciphering the data will be difficult, if not impossible, for most users. If you want to view the data contained in this key, you can do so using the System Information tool. To launch this tool go to Start, Help and Support, then click Support, then choose Advanced System Information from the See Also task items. Alternately, you can go to Start, Run, then type *msinfo32.exe* in the Open textbox or launch it from the Start menu (Start|All Programs|Accessories|System Tools|System Information).

HKEY_LOCAL_MACHINE\SAM

The subkey HKEY_LOCAL_MACHINE\SAM is a hive that contains data related to security. The **Security Accounts Manager (SAM)** database is stored in this key and is where local user accounts and group memberships are defined. The entire security structure of your Windows XP system is stored in this key. In most cases, these data are not accessible from a Registry editor, but instead reside in a file named SAM in the *%systemroot%*\WINDOWS\system32\config directory.

Caution

This is another area of the Registry that you should not normally attempt to modify. Most of the data contained in this subkey are in binary or encrypted format. You should employ the user manager tools (that is, the Local Users and Groups section of the Computer Management tool) to manipulate the data stored in this subkey. Additionally, to prevent you from editing it, this subkey has a security setting such that only the System (or the System utility) has rights to read and alter its contents.

HKEY_LOCAL_MACHINE\SECURITY

The subkey HKEY_LOCAL_MACHINE\SECURITY is the container for the local security policy, which defines control parameters, such as password policy, user rights, account lockout, audit policy, and general security options for the local machine. This subkey maps to a hive file named SECURITY in the *%systemroot%*\WINDOWS\system32\config directory.

Caution

This is yet another area of the Registry that you should not attempt to modify. Most of the data contained in this subkey are in binary format or are encrypted. You should employ the Local Security Policy tool to manipulate the data stored in this subkey (see Chapter 5, "Users, Groups, Profiles, and Policies"and 6, "Windows XP Security and Access Controls"). Additionally, to prevent you from editing this subkey, it has a security setting such that only the System has rights to read and alter its contents.

HKEY_LOCAL_MACHINE\SOFTWARE

The subkey HKEY_LOCAL_MACHINE\SOFTWARE is the container for data about installed software and mapped file extensions. These settings apply to all local users. The \Software\Classes subkey contains the same information as the HKEY_CLASSES_ROOT key; in fact the HKEY_CLASSES_ROOT key is created by copying the data from the \Software\Classes subkey. This subkey maps to a hive file named SECURITY in the *%systemroot%*\WINDOWS\system32\config directory.

HKEY_LOCAL_MACHINE\SYSTEM

The subkey HKEY_LOCAL_MACHINE\SYSTEM is the container for the information required to boot Windows XP. This subkey stores data about startup parameters, loading order for device drivers, service startup credentials (settings and parameters), and basic operating system behavior. This key is essential to the boot process of Windows XP. It contains subkeys called control sets that include complete information about the boot process for the system. This subkey resides in a hive file named "system" in the *%systemroot%*\WINDOWS\system32\config directory.

This subkey also contains additional subkeys with settings for storage devices (such as MountedDevices) and control set boot status (Select), and possibly subkeys left over from upgrading from Windows NT 4.0 (Disk and Setup). The control set keys are named and numbered; for example, ControlSet001 and ControlSet003. In most cases, there will be only two control sets numbered 001 and 003. These two sets represent the original (001) system configuration set and a backup (003) of the last functioning system configuration set. Thus, there will always be a functioning configuration to allow the operating system to boot (see Chapter 13, "Booting Windows XP").

Each control set has four subkeys (refer to Figure 12-1):

- *Control*—This is the container for data related to controlling system startup, boot parameters, computer name, and necessary subsystems to initiate.
- *Enum*—This is the container for data regarding required device drivers and their configuration.

- *Hardware Profiles*—This is the container for data specific to the hardware profile currently in use.
- *Services*—This is the container for data about drivers, services, file systems, applications, and other required hardware components necessary to load all installed and active services during bootup. This subkey also defines the order in which services are called and the way that one service can call or query other services.

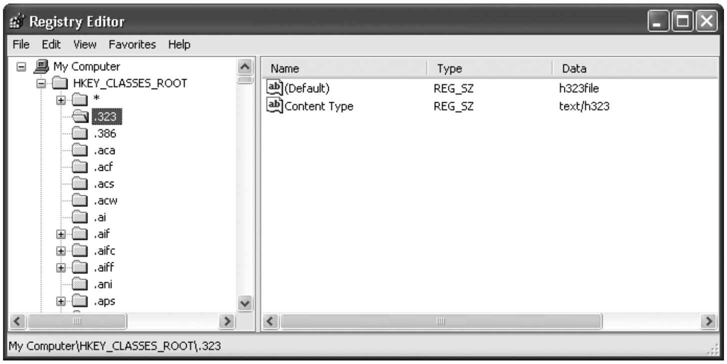
The value entries under the HKEY_LOCAL_MACHINE\SYSTEM\Select subkey are used to define how Windows XP uses its control. The four value entries are:

- *Default*—Defines which control set will be used during the next bootup
- *Current*—Lists the control set that was used to boot the current session
- *LastKnownGood*—Indicates the control set last used to boot and successfully log on a user (see later in this chapter for details and use)
- *Failed*—Lists the control set that was replaced by the control set from the LastKnownGood control set because of a failure to boot

The HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet subkey is a redirector to the actual ControlSet### currently in use rather than a truly distinct subkey. This symbolic link is used to simplify the programming interface for applications and device drivers that need information from the active control set. Because of this redirection, when you need to make modifications to the control set, you should use the CurrentControlSet "subkey" to direct your changes to the active control set properly.

HKEY_CLASSES_ROOT

The **HKEY_CLASSES_ROOT** key (see Figure 12-4) is the container for information pertaining to application associations based on file extensions and COM object data. The contents of this key are copied from the HKEY_LOCAL_MACHINE\SOFTWARE\Classes subkey. This key is maintained for backward compatibility with legacy applications and device drivers and is not strictly required by Windows XP.



```
My Computer
  HKEY_CLASSES_ROOT
    *
    .323
    .386
    .aca
    .acf
    .acs
    .acw
    .ai
    .aif
    .aifc
    .aiff
    .ani
    .aps
```

Name	Type	Data
(Default)	REG_SZ	h323file
Content Type	REG_SZ	text/h323

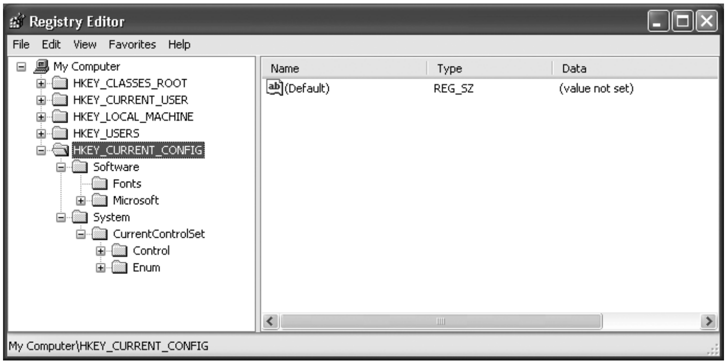
My Computer\HKEY_CLASSES_ROOT\.323

Figure 12-4 HKEY_CLASSES_ROOT contains file extension and com object settings and associations

As with other binary or protected keys, do not edit the contents of this key, or the HKEY_LOCAL_MACHINE\SOFTWARE\Classes subkey, directly. Instead, use the File Types tab of the Folder Options dialog box. To access this dialog box, select the Folder Options command from the Tools menu in Windows Explorer or My Computer or by launching the Folder Options applet from the Control Panel.

HKEY_CURRENT_CONFIG

The **HKEY_CURRENT_CONFIG** key (see Figure 12-5) is the container for data that pertain to whatever hardware profile is currently in use. This key is just a link to the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\HardwareProfiles\Current subkey. This key is maintained for backward compatibility with legacy applications and device drivers and is not strictly required by Windows XP.



```
My Computer
  HKEY_CLASSES_ROOT
  HKEY_CURRENT_USER
  HKEY_LOCAL_MACHINE
  HKEY_USERS
  HKEY_CURRENT_CONFIG
    Software
      Fonts
      Microsoft
    System
      CurrentControlSet
        Control
        Enum
```

Name	Type	Data
(Default)	REG_SZ	(value not set)

My Computer\HKEY_CURRENT_CONFIG

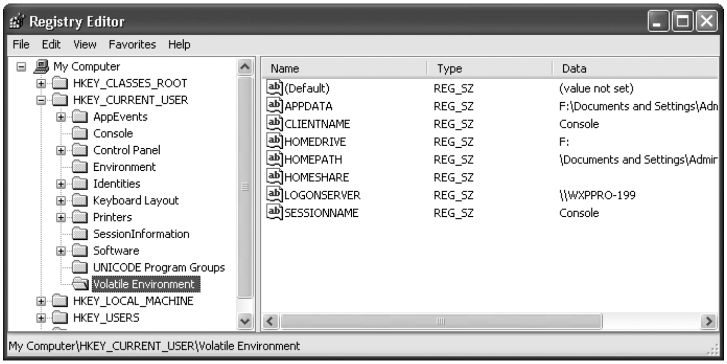
Figure 12-5 The HKEY_CURRENT_CONFIG/ key is maintained in Windows XP for backward compatibility

Caution

The contents of this key, and the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware Profiles\Current subkey, should not be edited directly. Instead, the Hardware Profiles interface or Device Manager should be used. The Hardware Profiles interface is accessed by pressing the Hardware Profiles button on the Hardware tab of the System applet from the Control Panel. The Device Manager is accessed by pressing the Device Manager button on the Hardware tab of the System applet from the Control Panel or by selecting the Device Manager node from the Computer Management utility in Administrative Tools.

HKEY_CURRENT_USER

The **HKEY_CURRENT_USER** key (see Figure 12-6) is the container for the profile for whichever user is currently logged on. The contents of this key are built each time a user logs on by copying the appropriate subkey from the HKEY_USERS key. The contents of this key should not be edited directly; instead, you should modify a user's profile through conventional profile management techniques (see Chapter 5 for more information on profile management).



- My Computer
  - HKEY_CLASSES_ROOT
  - HKEY_CURRENT_USER
    - AppEvents
    - Console
    - Control Panel
    - Environment
    - Identities
    - Keyboard Layout
    - Printers
    - SessionInformation
    - Software
    - UNICODE Program Groups
    - Volatile Environment
  - HKEY_LOCAL_MACHINE
  - HKEY_USERS

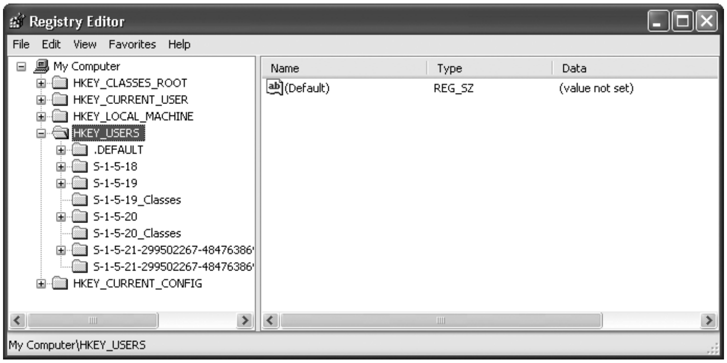
Name	Type	Data
(Default)	REG_SZ	(value not set)
APPDATA	REG_SZ	F:\Documents and Settings\Adn
CLIENTNAME	REG_SZ	Console
HOMEDRIVE	REG_SZ	F:
HOMEPATH	REG_SZ	\Documents and Settings\Admir
HOMESHARE	REG_SZ	
LOGONSERVER	REG_SZ	\\WXPPRO-199
SESSIONNAME	REG_SZ	Console

My Computer\HKEY_CURRENT_USER\Volatile Environment

Figure 12-6 HKEY_CURRENT_USER contains data for whichever user is currently logged onto the system

HKEY_USERS

The **HKEY_USERS** key (see Figure 12-7) contains profiles for all users who have ever logged onto this system and the default user profile. The contents of this key are built each time the system boots by loading the default file and the locally stored copies of Ntuser.dat or Ntuser.man from user profiles (see Chapter 5). These locally stored copies are found in the \Documents and Settings\<*username*> directory on a Windows XP Professional system. To remove a user profile from this key, use the User Profiles tab of the System applet from the Control Panel. To alter the contents of a profile, use conventional profile management techniques (see Chapter 5 for more information on profile management) instead of attempting to edit this key directly. Note also that subkeys in this key use Windows Security IDs (SIDs) to identify users, rather than account names, which explains their cryptic alphanumeric names.



Name	Type	Data
(Default)	REG_SZ	(value not set)

My Computer\HKEY_USERS

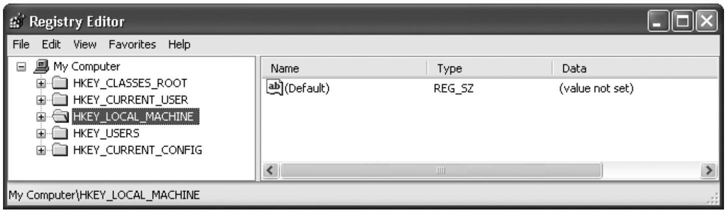
Figure 12-7 HKEY_USERS contains data for any user who's ever logged onto the system, plus a default user profile

HKEY_DYN_DATA

In some Registries, you may occasionally run across another main key named HKEY_DYN_DATA. This root or main key appears only on machines with Windows 95 or Windows 98 applications that use older versions of Plug and Play to detect and track hardware devices as they enter or leave a system. Because Windows XP Professional's Plug and Play implementation is vastly superior to these older versions, this entry exists solely to help the operating system maintain backward compatibility with older versions of Windows.

Registry Editors

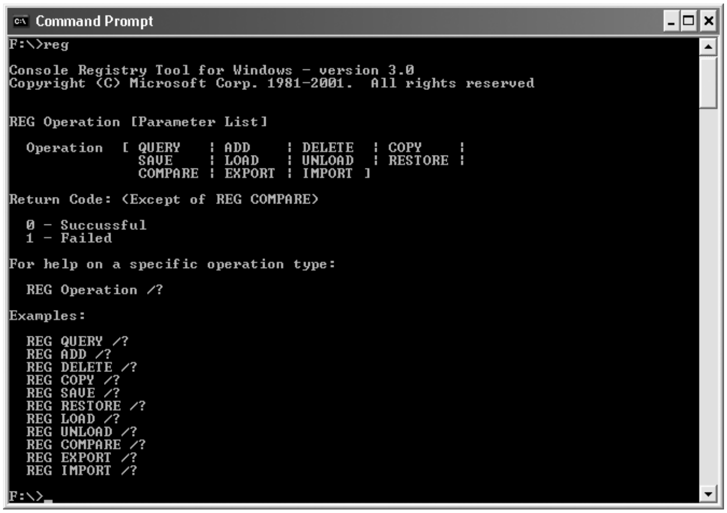
Because the structure of the Registry is so complex, special tools are required to operate on it directly. The primary Registry editor for Windows XP is launched by executing either Regedit.exe or Reg.exe. **Regedit** (see Figure 12-8) offers global searching, security manipulation, and combines all of the keys into a single display. **Reg** (see Figure 12-9) is the Console Registry Tool for Windows, a command-line utility that permits users, batch files, or programs to operate on the Registry, but that supports no attractive graphical user interface like that for Regedit.



Name	Type	Data
(Default)	REG_SZ	(value not set)

My Computer\HKEY_LOCAL_MACHINE

Figure 12-8 Regedit is the older Registry editor that suffices for most uses



```
F:\>reg

Console Registry Tool for Windows - version 3.0
Copyright (C) Microsoft Corp. 1981-2001.  All rights reserved


REG Operation [Parameter List]

  Operation  [ QUERY   | ADD    | DELETE  | COPY    |
               SAVE    | LOAD   | UNLOAD  | RESTORE |
               COMPARE | EXPORT | IMPORT ]

Return Code: (Except of REG COMPARE)

  0 - Successful
  1 - Failed

For help on a specific operation type:

  REG Operation /?

Examples:

  REG QUERY /?
  REG ADD /?
  REG DELETE /?
  REG COPY /?
  REG SAVE /?
  REG RESTORE /?
  REG LOAD /?
  REG UNLOAD /?
  REG COMPARE /?
  REG EXPORT /?
  REG IMPORT /?

F:\>_
```

Figure 12-9 Reg.exe is a command-line utility that permits users, batch files, or programs to operate on the Registry

Because it is a command-line utility and does not display the Registry's hierarchical organization in an easy-to-grasp form, Reg.exe is not as convenient or friendly as Regedit.exe. However, both editors can be used to view keys and values (see Hands-on Project 12-1), perform searches (see Hands-on Project 12-2), add new subkeys and value entries, alter the data in value entries, and import and export keys and subkeys. For most purposes, however, Regedit should be your primary Registry inspection and editing tool.

Caution

As already noted many times in this chapter, editing the Registry directly should not be undertaken without forethought and planning. It is possible to alter the Registry, whether on purpose or accidentally, in such a way as to render a system completely unrecoverable. If you don't know exactly what you are doing, *don't do it!* Please also note that although earlier versions of Windows included another GUI Registry editor called Regedt32.exe, that program is no longer available as part of Windows XP Professional. However, executing Regedit32 launches the existing Registry editor tool.

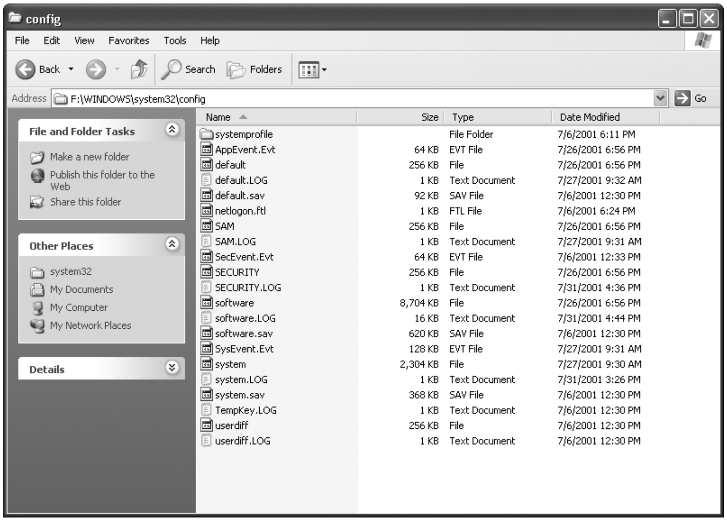
Even when you do think you know exactly what you want to change in the Registry, it is always a good idea to take precautions, such as the following:

- Back up all important data on the computer before editing the Registry.
- Make a distinct backup of all or part of the Registry. Saving each key or subkey individually is recommended (see Hands-on Project 12-3). Saving parts of the Registry to files enables you to restore parts of the Registry instead of the entire Registry. Store the backup files on local drives, network drives, and floppies or other removable media to ensure access.

- Reboot the machine before editing the Registry.
- Perform only a single Registry modification at a time. Test the results before proceeding.
- Reboot immediately after each change to force full system compliance with new settings in the Registry. This is not strictly necessary, but has often proved to be prudent.
- Always test changes on a nonproduction system hosting noncritical services before deploying on production systems.

Registry Storage Files

The files in which a static image of the Registry are stored reside in the *%systemroot%*\WINDOWS\system32\config and *%systemroot%*\WINDOWS\repair directories of the boot partition (see Figure 12-10). The Registry is not stored in files that match one-to-one with the top-level keys, as we explain shortly, but there are plenty of Registry data mapped into files for safekeeping (and to maintain backup or rollback versions of these data).



Address: F:\WINDOWS\system32\config

Name	Size	Type	Date Modified
systemprofile		File Folder	7/6/2001 6:11 PM
AppEvent.Evt	64 KB	EVT File	7/26/2001 6:56 PM
default	256 KB	File	7/26/2001 6:56 PM
default.LOG	1 KB	Text Document	7/27/2001 9:32 AM
default.sav	92 KB	SAV File	7/6/2001 12:30 PM
netlogon.ftl	1 KB	FTL File	7/6/2001 6:24 PM
SAM	256 KB	File	7/26/2001 6:56 PM
SAM.LOG	1 KB	Text Document	7/27/2001 9:31 AM
SecEvent.Evt	64 KB	EVT File	7/6/2001 12:33 PM
SECURITY	256 KB	File	7/26/2001 6:56 PM
SECURITY.LOG	1 KB	Text Document	7/31/2001 4:36 PM
software	8,704 KB	File	7/26/2001 6:56 PM
software.LOG	16 KB	Text Document	7/31/2001 4:44 PM
software.sav	620 KB	SAV File	7/6/2001 12:30 PM
SysEvent.Evt	128 KB	EVT File	7/27/2001 9:31 AM
system	2,304 KB	File	7/27/2001 9:30 AM
system.LOG	1 KB	Text Document	7/31/2001 3:26 PM
system.sav	368 KB	SAV File	7/6/2001 12:30 PM
TempKey.LOG	1 KB	Text Document	7/6/2001 12:30 PM
userdiff	256 KB	File	7/6/2001 12:30 PM
userdiff.LOG	1 KB	Text Document	7/6/2001 12:30 PM

Figure 12-10 Explorer listing of the ...\system32\config folder shows various Registry file types and instances

The Registry is stored in various subkey, logging, and backup files, as indicated in Table 12-1.

Table 12-1 Registry Storage Files

Registry Key/Subkey	Storage Files
HKEY_LOCAL_MACHINE\SAM	Sam, Sam.log, Sam.sav
HKEY_LOCAL_MACHINE\SECURITY	Security, Security.log, Security.sav
HKEY_LOCAL_MACHINE\SOFTWARE	Software, Software.log, Software.sav
HKEY_LOCAL_MACHINE\SYSTEM	System, System.alt, System.log, System.sav
HKEY_USERS\.DEFAULT	Default, Default.log, Default.sav
(Not directly associated with a Registry key)	Userdiff, Userdiff.log
HKEY_CURRENT_USER	Ntuser.dat, Ntuser.dat.log

> Note
>
> Note that only four of the HKEY_LOCAL_MACHINE subkeys, the Default subkey of the HKEY_USERS key, and the HKEY_CURRENT_USER key are stored in files. All of the other keys and subkeys are either built "on the fly" at bootup or are copies of a subsection of HKEY_LOCAL_MACHINE.

The HKEY_USERS key is built from the default file (which represents the default user profile's NTUSER.DAT file) and copies of profiles for all users who have ever logged onto the computer. These profiles are cached locally in the \Documents and Settings\<*username*> directory. A copy of the NTUSER.DAT or Ntuser.man file is copied into the repair directory for the currently logged-on user.

Notice that four extensions are used by the Registry storage files to identify the purpose or function of the file:

- *No extension*—The storage file for the subkey itself, also known as a hive file.
- *.alt*—The backup file for the subkey. Note that only the HKEY_LOCAL_MACHINE\SYSTEM subkey has a backup file.
- *.log*—A file containing all changes made to a key. This file is used to verify that all modifications to the Registry are properly applied.
- *.sav*—Copies of keys in their original state as created at the end of the text portion of Windows XP installation.

> Note
>
> TechNet now includes a wonderful *Windows NT Magazine* article entitled "Inside the Registry" from Mark Russinovitch, a leading Windows expert. Online, you can find this article at www.microsoft.com/technet/treeview/default.asp?url=/TechNet/prodtechnol/winntas/tips/winntmag/inreg.asp. Note that even though the article is somewhat outdated, it's still a worthwhile read.

Under Windows NT 4.0, the Registry files stored in the \Config directory were used to build the emergency repair disk (ERD). Under Windows 2000 and XP, these files are no longer copied onto the ERD when it is created. However, you can create your own custom ERD by manually copying the files in the \Config directory to a formatted floppy. You may find having a complete copy of the Registry quite handy when you need to perform a system repair or restore any portion of the Registry because of corruption or human error. If you need to use those files, you can always use the Import command in Regedit to restore that data to a damaged Registry.

REGISTRY FAULT TOLERANCE

If the Registry becomes corrupted or destroyed, Windows XP cannot function or even boot. Several mechanisms have been established to prevent the Registry from becoming damaged or to repair minor problems automatically. The fault tolerance of the Registry is sustained by its structure, memory residence, and transaction logs. These mechanisms ensure that all changes or operations performed on the Registry either succeed or fail. This prevents any partially applied alterations that would result in an invalid value entry or entries. Thus an "all or nothing" guarantee is supported no matter what method of alteration is used, including using a Registry editor or an administrative tool or alterations by an application. If the change action is interrupted (by power failure, too little CPU time, hardware failure, etc.), the Registry remains intact, even if the desired change was not implemented.

As previously mentioned, when a value entry is altered in the Registry, that change applies to the copy of the Registry stored in active memory. This means that the change affects the system immediately in most cases. A change to the Registry is only made permanent when key files are copied back to the hard drive. This activity occurs during a **flush**, a copy procedure to update the files on the hard drive with the new settings stored in the memory-resident version of the Registry. A flush occurs at shutdown, when forced by an application, or just after a Registry alteration.

Transaction logs are files wherein the system records edits, changes, and alterations to the Registry, similar to a list of orders or commands. When a flush occurs, the transaction log is updated to record all changes currently in memory, which will be written to the Registry storage files. This log is used by the system to verify automatically that all Registry changes are correct as the flush concludes.

A flush includes the following sequence of steps:

1. All alterations to a key are appended to that key's transaction log file (.log).
2. The key file is marked as being in transition.
3. The key file is updated with the new data from memory.
4. The key file is marked as complete.

If a system failure occurs between the time that the key file is marked as in transition and when it is marked complete, the original state of the key is recovered using the data from the transaction log. If the flush finishes uninterrupted, the system continues to perform normally.

The flush operation is performed on all keys except the SYSTEM subkey. This subkey contains system-critical data and is a major ingredient in a successful boot-up of Windows XP. For this reason, recovery cannot rely upon transaction logs. Instead, Windows XP updates the SYSTEM subkey using a different method:

1. The system file is marked as being in transition.
2. The system file is brought up to date with the state of the Registry from memory.
3. The system file is marked as being complete.
4. The System.alt file is marked as being in transition.
5. The System.alt file is brought up to date with the state of the Registry from memory.
6. The System.alt file is marked as being complete.

This dual-file process, with its primary and backup copies of the SYSTEM subkey file, ensures that no matter at which stage the update process might be interrupted, a complete and functional copy of the SYSTEM subkey file is available. If the failure occurs within the first three steps, the nonupdated System.alt file is used to boot. If the failure occurs within the last three steps, the updated system file is used to boot. Once booting is complete after a failure, Windows XP performs the update again to ensure that both copies of the SYSTEM subkey are exactly the same. However, if the failure occurs during the first three steps, any changes made to the system will have been lost.

Though Windows XP automatically manages the safety of the Registry through its fault-tolerance mechanisms (.log and .alt files), it is still important for you to take proactive measures to back up the Registry. There are several ways to create reliable Registry backups:

- Most Windows XP backup applications (for example, the built-in Backup tool and third-party products such as Veritas Backup Exec and Stac Replica) include support for full Registry backups. With these products, you can back up the Registry as part of your daily automated backup or as a distinct Registry-only procedure. Backing up the Registry with most of these products consists of selecting a "Back up the Registry" or "System State" check box when you make file/folder selections before initializing a backup.
- Regedit can be used to save all or part of the Registry to distinct files. This tool offers an Export command, which may be used to save the entire Registry, a single key, or any subportion of a key to a file (try Hands-on Project 12-3).
- Make a copy of the *%systemroot%*\WINDOWS\system32\config and *%systemroot%*\WINDOWS\repair directories manually. Just copy the contents to another location on your local computer, on a drive elsewhere on your network, or to a floppy disk (if size allows) or recordable CD.

- Employ the *Microsoft Windows XP Professional Resource Kit* tools, such as Regback.exe.This tool offers command-line scripting capabilities. Explore the *Microsoft Windows XP Professional Resource Kit* for ideas on how to best employ these tools.You can see a syntax parameter listing for these and most command-line tools by issuing a "/?" parameter after the command from a Command Prompt (that is, *reg /?*, or *reg /? | more* if more than one screen's worth of data is displayed).

Note

No matter which backup method you employ, take the time to make two copies or perform the backup twice. This provides additional insurance in case your first backup fails.

Restoring the Registry

Obviously, if you are going to take the time to create backups of the Registry, you must understand how to restore it.You have several options for restoring the Registry, depending on the method used to make a backup. Windows XP itself attempts to maintain a functional Registry, using its own internal automatic fault-tolerance mechanisms. If the automatic restoration process fails, you can first attempt to restore the Last Known Good Configuration. The **Last Known Good Configuration (LKGC)** is the state of the Registry stored in one of the control sets (covered earlier this chapter) when the last successful user logon occurred. If the Registry is damaged in such a way that it cannot fully boot or won't allow a user to log on, the LKGC option can restore the system to its prior working state.

This boot option is accessed by pressing F8 during the initial bootup of Windows XP when the boot menu is displayed. Don't worry; the basic boot menu even prompts you to press F8 if you need an alternate boot method. Pressing F8 reveals a new selection menu similar to the following:

```
Windows Advanced Options Menu
Please select an option:

Safe Mode
Safe Mode with Networking
Safe Mode with Command Prompt

Enable Boot logging
Enable VGA Mode
Last Known Good Configuration (your most recent settings that worked)
Directory Services Restore Mode (Windows domain controllers only)
```

```
Debugging Mode


Start Windows Normally
Reboot
Return to OS Choices Menu


Use the up and down arrow keys to move the highlight to your choice.
```

Use the arrow keys to highlight the Last Known Good Configuration selection, then press Enter. Keep in mind that any changes made to the system between the time the LKGC was stored and its use to restore the system will be lost. If the LKGC fails to restore normal system functions, you have only two options:

1. Use your backup software to restore the Registry files. This is only possible if your backup application offers a DOS-based restore mechanism that can bypass NTFS write restrictions. In other words, the backup software must operate without a functional Windows XP environment when launched from a bootable floppy. This type of software lets you restore files to the boot and system partitions (such as the Registry) so you can return to a functional OS. Unfortunately, these applications are few and far between. One such product is Replica from Stac (*www.stac.com*).
2. Reinstall Windows XP, either fully or as an upgrade. An upgrade may replace the section of the Registry that is causing the problems, allowing you to retain most of your configuration, but this is not guaranteed. A full, new installation of Windows XP will return the system to a preconfigured state and require you to repeat all post-installation changes you may have made.

If you are able to boot into the system, but things are not functioning the way they should; or if services, drivers, or applications are not loading or operating properly, you may need to restore the Registry in part or whole from backup. Simply use the same tool employed to create the backup to restore the Registry. Keep in mind that some tools allow you to restore portions of the Registry instead of the whole thing (see Hands-on Project 12-4).

No matter what method you employ to restore the Registry, it's always a good idea to reboot the system to ensure that the restore operation completed successfully and that the system is using only working (or more correctly, reverted-to) settings. It's also a good idea to retain the copies of the old Registry until you are confident that the system is functioning normally and have had the opportunity to create new backups. In other words, don't throw away the disks, erase the drives, or format the tapes containing the Registry backup; keep a few generations of Registry backups on hand, just in case.

Windows XP Professional Resource Kit Registry Tools

The *Microsoft Windows XP Professional Resource Kit* includes several tools that can be used to manipulate the Registry. The *Microsoft Windows XP Professional Resource Kit* is a Microsoft product separate from the Windows XP Professional operating system. The *Microsoft Windows XP Professional Resource Kit* has additional documentation on Windows XP Professional, its operations, and its use, as well as a host of useful tools and utilities not included with the standard operating system software. You can purchase the *Microsoft Windows XP Professional Resource Kit* from Microsoft or from most software or book vendors.

Because many of these tools are command-line tools or require perusal of significant ancillary materials, we recommend that you read over the *Microsoft Windows XP Professional Resource Kit* documentation yourself before actually using these tools. Some of the key utilities include:

- *Regdump.exe*—A command-line tool used to dump all or part of the Registry to Stdout (this is an abbreviation for the standard output file, where the system creates output by default; normally this sends the output to a file whose name you specify when you run the command). The output of this tool is suitable for the Regini.exe tool. This tool is useful when you need to create scripts based on Registry content by creating a dump of existing settings.
- *Regfind.exe*—A command-line tool used to search the Registry for a key, value name, or value data based on keywords.
- *Compreg.exe*—A GUI tool used to compare two local or remote Registry keys and highlight all differences.
- *Regini.exe*—A command-line scripting tool used to add keys into the Registry.
- *Regback.exe*—A command-line scripting tool used to back up keys from the Registry.
- *Regrest.exe*—Another command-line scripting tool used to restore keys to the Registry.
- *Scanreg.exe*—A GUI tool used to search the Registry for a key, value name, or value data based on keywords.

Chapter Summary

- The Windows XP Registry is a complex structure consisting of keys, subkeys, values, and value entries.
- The Registry should be manipulated with extreme caution. Unless absolutely necessary, the Registry should not be edited directly; instead, employ the Control Panel applets and Administration Tools to modify system settings.

- Windows XP maintains a functional Registry through several fault-tolerant measures, including transaction logs and backup of key files.
- The Registry is divided into five main keys. The primary and most important key is HKEY_LOCAL_MACHINE, because it hosts data ranging from system startup information to driver settings to the security database. For some, but not all, of these main keys, Windows XP Professional writes them to files in the *%systemroot%*\WINDOWS\system32\config directory that are called hives or hive files.
- Windows XP includes two Registry editors, the graphical Regedit.exe and the command-line Reg.exe utility. The former is useful for global searches and general inspection or quick edits, the latter for performing systematic or comprehensive user- or program-driven Registry edits.
- As part of your normal system maintenance and administration, you should create copies of the Registry. Backing up the Registry often is the only way to ensure you have a functional Registry to restore in the event of a failure.

Key Terms

Binary — A Registry value entry data type that stores data in binary format.

DWORD — A Registry value entry data type that stores data in binary, hex, or decimal format.

data type — The setting on a Registry value entry that defines the data format of the stored information.

Expandable String — A Registry value entry data type that stores data in expandable text-string format containing a variable that is replaced by an application when used (for example, *%systemroot%*\FILE.EXE).

flush — Forcing the memory-resident copy of the Registry to be written to files stored on the hard drive. A flush occurs at shutdown, when forced by an application, or just after a Registry alteration.

hive — A discrete body of Registry keys, subkeys, and values stored in a file.

HKEY_CLASSES_ROOT — This Registry key contains the value entries that control the relationships between file extensions (and therefore file format types) and applications. This key also supports the data used in object linking and embedding (OLE), COM object data, and file-class association data. This key actually points to another Registry subkey named HKEY_LOCAL_MACHINE\SOFTWARE\Classes and provides multiple points of access to make itself easily accessible to the operating system itself and to applications that need access to the compatibility information already mentioned.

HKEY_CURRENT_CONFIG — This Registry key contains the value entries that control the currently active hardware profile; its contents are built each time the system is booted. This key is derived from data stored in the HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\HardwareProfiles\Current subkey. HKEY_CURRENT_CONFIG exists to provide backward-compatibility with Windows 95/98 applications.

HKEY_CURRENT_USER — This Registry key contains the value entries that define the user environment for the currently logged-on user. This key is built each time a user logs on to the system. The data in this key are derived from the HKEY_USERS key and the NTUSER.DAT and NTUSER.MAN files of a user's profile.

HKEY_LOCAL_MACHINE — This Registry key contains the value entries that control the local computer. This includes hardware devices, device drivers, and various operating system components. The data stored in this key are not dependent on a logged-on user or the applications or processes in use.

HKEY_USERS — This Registry key contains the value entries that define the user environments for all users who have ever logged on to this computer. As a new user logs on to this system, a new subkey is added for that user that is built either from the default profile stored in this key or from the roaming user profile associated with the domain user account.

key — A top-level division of the Registry. There are five keys in a Windows XP Registry. A key can contain subkeys.

Last Known Good Configuration (LKGC) — The state of the Registry stored in one of the control sets when the last successful user logon was performed. If the Registry is damaged in such a way that it will not fully boot or will not allow a user to log on, the LKGC option can restore the system to a previous state. Keep in mind that any changes made to the system between the time the LKGC was stored and its use to restore the system will be lost.

Multiple String — A Registry value entry data type that stores data in text-string format containing multiple human-readable values separated by null characters.

Reg — A special command-line utility that users, programs, or the operating system can use to access, inspect, create, or modify Registry keys.

Regedit — The 16-bit Registry editor. Regedit offers global searching and combines all of the keys into a single display. It can be used to perform searches, add new subkeys and value entries, alter the data in value entries, and import and export keys and subkeys.

Registry — The hierarchical database of system configuration data essential to the health and operation of a Windows XP system.

Security Accounts Manager (SAM) — The database of user accounts, group memberships, and security-related settings.

String — A Registry value entry data type that stores data in text-string format.

subkey — A division of a Registry key, such as HKEY_LOCAL_MACHINE. A subkey can contain other subkeys and value entries.

transaction log — A file created by Windows XP to record Registry changes. These files, with a .log extension, are used to verify that changes to the Registry are made successfully.

value — The actual data stored by a value entry.

value entry — A named Registry variable that stores a specific value or data string. A Registry value entry's name is typically a multiword phrase without spaces and with title capitalization.

Review Questions

1. The Registry is the primary mechanism for storing data about Windows XP. Which of the following are configuration files used by other Microsoft operating systems and may still exist on Windows XP for backward compatibility? (Choose all that apply.)
   a. Win.ini
   b. Autoexec.bat
   c. System.ini
   d. Config.sys
2. The Registry is only used to store configuration data for native Windows 2000 applications, services, and drivers. True or False?
3. Which of the following tools is most highly recommended by Microsoft for editing the Registry? (Chose all that apply.)
   a. Control Panel applets
   b. Regedit
   c. Reg.exe
   d. Administrative Tools
4. The Registry is an exhaustive collection of system control parameters. True or False?
5. When editing the Registry, especially when attempting to alter the unseen defaults, which of the following pieces of information are important? (Choose all that apply.)
   a. syntax
   b. spelling
   c. subkey location
   d. valid values
   e. time zone
6. Changes made to the Registry never go into effect until the system is rebooted. True or False?
7. The Windows XP Professional Registry has how many default keys?
   a. 2
   b. 4
   c. 5
   d. 6

8. Which of the following can host subkeys or values?
   a. data type
   b. key
   c. subkey
   d. value data
9. Each of the highest-level keys of the Registry are stored in a distinct file on the hard drive. True or False?
10. Which Registry key contains the value entries that control the local computer?
   a. HKEY_LOCAL_MACHINE
   b. HKEY_CLASSES_ROOT
   c. HKEY_CURRENT_CONFIG
   d. HKEY_USERS
11. Which Registry key contains the value entries that define the user environment for the currently logged-on user?
   a. HKEY_LOCAL_MACHINE
   b. HKEY_CLASSES_ROOT
   c. HKEY_CURRENT_CONFIG
   d. HKEY_CURRENT_USER
12. Which Registry key contains the value entries that control the relationships between file extensions (and therefore file format types) and applications?
   a. HKEY_LOCAL_MACHINE
   b. HKEY_CLASSES_ROOT
   c. HKEY_CURRENT_CONFIG
   d. HKEY_USERS
13. Which Registry key contains the value entries that control the currently active hardware profile?
   a. HKEY_LOCAL_MACHINE
   b. HKEY_CLASSES_ROOT
   c. HKEY_CURRENT_CONFIG
   d. HKEY_CURRENT_USER
14. From which key can you delete subkeys, using the System applet?
   a. HKEY_LOCAL_MACHINE
   b. HKEY_CLASSES_ROOT
   c. HKEY_CURRENT_CONFIG
   d. HKEY_USERS

15. Some Windows 95 or 98 applications require a sixth Registry key. Windows XP adds the ________ key, which is actually a redirector rather than an actual key, to maintain backward compatibility.
16. After you've created a value entry, you can easily change its data type by using the Edit dialog box. True or False?
17. The value entry data type that can store binary, hex, or decimal formatted data is:
    a. String
    b. DWORD
    c. Multi-String
    d. Expandable String
18. Where are the files used to load the Registry at bootup stored on a Windows XP system?
    a. *%systemroot%*\config
    b. *%systemroot%*\system32\config
    c. *%systemroot%*\system\config
    d. *%systemroot%*\system32\repair
19. Which subkey of HKEY_LOCAL_MACHINE is the only subkey to have a backup file?
    a. SAM
    b. SOFTWARE
    c. SYSTEM
    d. SECURITY
20. The process of pushing Registry changes from memory to a hard drive file is known as ________________.
21. Which type of file (specified by file extension) does Windows XP use to record the changes to the Registry for verification purposes?
    a. .alt
    b. .sav
    c. .dat
    d. .log
22. Assume that your system is performing an update to the SYSTEM subkey. While altering the system file, before working on the System.alt file, a system crash occurs. When the system reboots, which of the following will occur?
    a. You'll be prompted whether to use the system or System.alt set of configuration parameters.
    b. The state of the Registry before changes to the SYSTEM subkey will be restored.
    c. The state of the Registry after changes to the SYSTEM subkey will be restored.
    d. The system will fail to boot because of a corrupt SYSTEM subkey.

23. Which subkey usually cannot be edited with a Registry editor?
    a. HARDWARE
    b. SOFTWARE
    c. SAM
    d. CurrentControlSet
24. Which control set subkey is the container for data related to controlling system startup, boot parameters, computer name, and necessary subsystems to initiate?
    a. Control
    b. Enum
    c. Hardware Profiles
    d. Services
25. Which subkey of HKEY_LOCAL_MACHINE\SYSTEM\Select indicates the control set that was last used to boot and successfully log on a user?
    a. Default
    b. Current
    c. LastKnownGood
    d. Failed

HANDS-ON PROJECTS

Project 12-1

This hands-on project presents the necessary steps in viewing the current value of a value entry or determining if a value entry is even present in the Registry.

To view Registry value entries with Regedit:

1. Select **Start|Run**.
2. Type **regedit**, then click **OK**. The Registry Editor opens.
3. Double-click **HKEY_LOCAL_MACHINE**.
4. Locate and double-click **SOFTWARE** under HKEY_LOCAL_MACHINE.
5. Locate and double-click **Microsoft** under SOFTWARE.
6. Locate and double-click **Windows NT** under Microsoft.
7. Locate and double-click **CurrentVersion** under Windows NT.
8. Locate and select **Winlogon** under CurrentVersion.
9. In the right pane, locate and select **DefaultUserName**.

10. From the Edit menu, select **Modify**.
11. Notice that the value of this value entry is the name of your current user account.
12. Click **Cancel**.
13. In the left pane, scroll up until you see HKEY_LOCAL_MACHINE.
14. Double-click **HKEY_LOCAL_MACHINE**. Leave the system as is for the next hands-on project.

Hands-on Project

Project 12-2

To search for a value entry with Regedit:

Note

This hands-on project requires that Hands-on Project 12-1 be completed. In this project, you use Regedit to locate a key or value without knowing its path within the Registry. You will begin at the system status point where the previous hands-on project ended.

1. From the **Edit** menu, select **Find**.
2. In the Find what field, type **DefaultUserName**.
3. Click **Find Next**. Regedit locates the first key, value, or data containing that string.
4. Notice that the first found match is AltDefaultUserName (be patient; this first search may take as long as a minute to complete).
5. From the Edit menu, select **Find Next** (or click the hotkey equivalent, F3).This is the actual DefaultUserName value entry that you viewed in Hands-on Project 12-1.
6. In the left pane, scroll up until you see HKEY_LOCAL_MACHINE.
7. Double-click **HKEY_LOCAL_MACHINE**. Leave the system as is for the next hands-on project.

Hands-on Project

Project 12-3

The ability to make backups of the Registry offers you an additional level of support in the event of a system problem or a human error in regard to the Registry. Plus, backing up the Registry is always a good idea before beginning any modifications to the Registry, either through the manual tool in Regedit or through any of the Control Panel or Administrative Tools utilities.

To back up a Registry key:

Note

This hands-on project begins at the system status point where the previous hands-on project ended.

1. Make sure that the HKEY_USERS key is selected.

2. From the **File** menu, select **Export**.
3. Select a destination folder (such as c:\temp) of your choice in the **Save in:** pull-down menu.
4. Provide a file name, such as **HKUsave.reg**.
5. Make sure the **Selected branch** radio button at the bottom of the Export Registry File dialog box is selected and that HKEY_USERS is listed in the text field.
6. Click **Save**.
7. The Regedit tool will create a backup file of the selected key. Leave the system as is for the next hands-on project.

This procedure can be used to back up the entire Registry or just a small subset of subkeys, simply by selecting different keys or subkeys.

Project 12-4

If you have made any changes to the system or Registry since Hands-on Project 12-3 was performed, you may not want to perform this project, because it will discard those changes by restoring the state of the Registry from the saved file.

To restore a Registry key:

This hands-on project requires that Hands-on Project 12-3 be completed. It begins at the system status point where Hands-on Project 12-3 ended.

1. From the **File** menu, select **Import**.
2. Locate and select your **HKUsave.reg** file.
3. Click **Open**.
4. After a few moments of importing, a message stating whether the import succeeded is displayed; click **OK**.
5. From the **File** menu, select **Exit**.

As with backing up a Registry key, this procedure can be used to restore the entire Registry or just a small subset of subkeys simply by selecting different keys or subkeys. However, it does require that the same amount or even more data be backed up for the material to be restored.

Project 12-5

To use Reg.exe, the Windows Console Registry Tool:

1. Select **Start**|**All Programs**|**Accessories**|**Command Prompt**.

2. Type **reg** to produce the basic documentation for the utility.
3. Notice that each major key may be abbreviated for compactness.
4. Type **reg query "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon" /v DefaultUserName** to display the contents of that value entry (it should match your current logon name). Press **Enter**.

Project 12-6

To view security permissions with Regedit:

1. Launch Regedit (**Start**, **Run**, then type **Regedit** into the Open: textbox).
2. From the Window menu, select **HKEY_USERS**.
3. From the **Edit** menu, select **Permissions**.
4. Notice the Permissions dialog box for the Registry is identical to that used elsewhere in Windows XP.
5. Click **Cancel**.
6. From the **File** menu, select **Exit**.

Case Projects

1. Describe the actions that you can perform manually or that are performed automatically to provide protection or fault-tolerance mechanisms for the Windows XP Registry.
2. You have been asked to perform several Registry modifications to fine-tune an application. You'll be following detailed instructions from the vendor. What steps can you take to ensure that even if the vendor's instructions fail, you'll be able to return to a functioning Windows XP system?